

Muon tagger: NSB-scan efficiency

08.10.2026



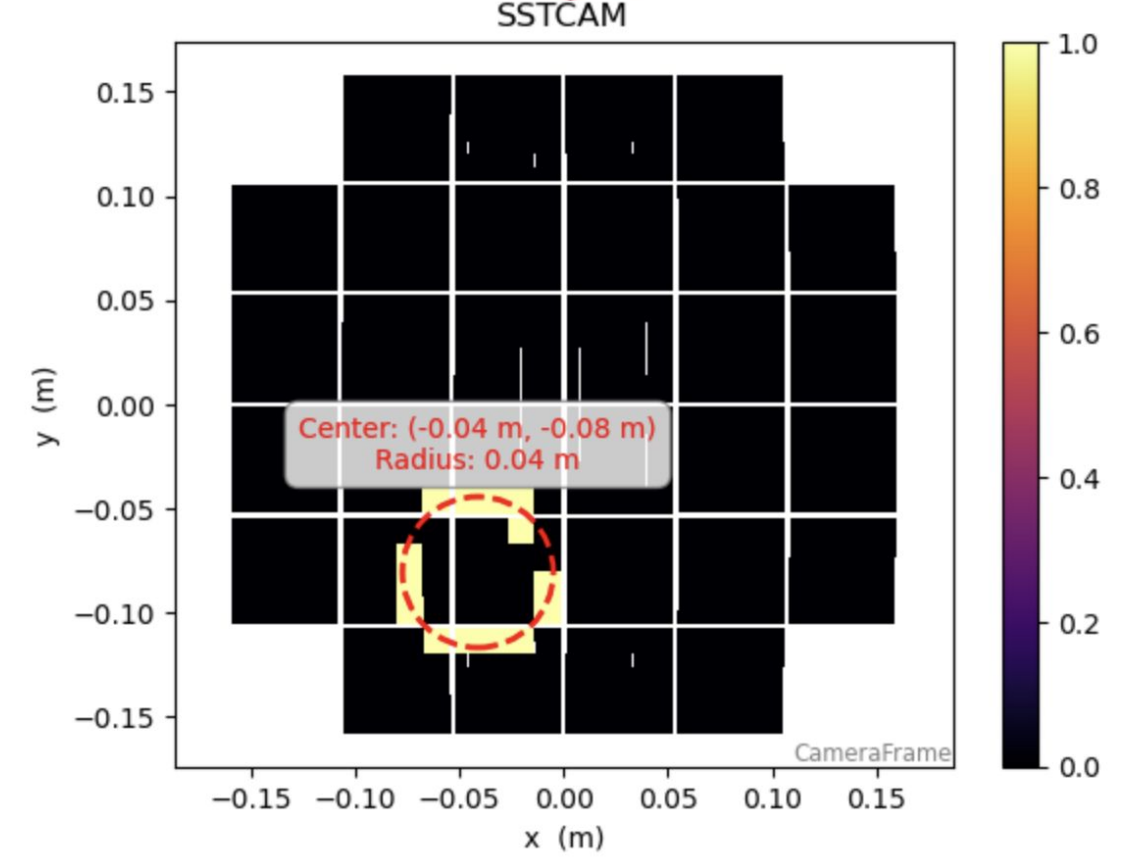
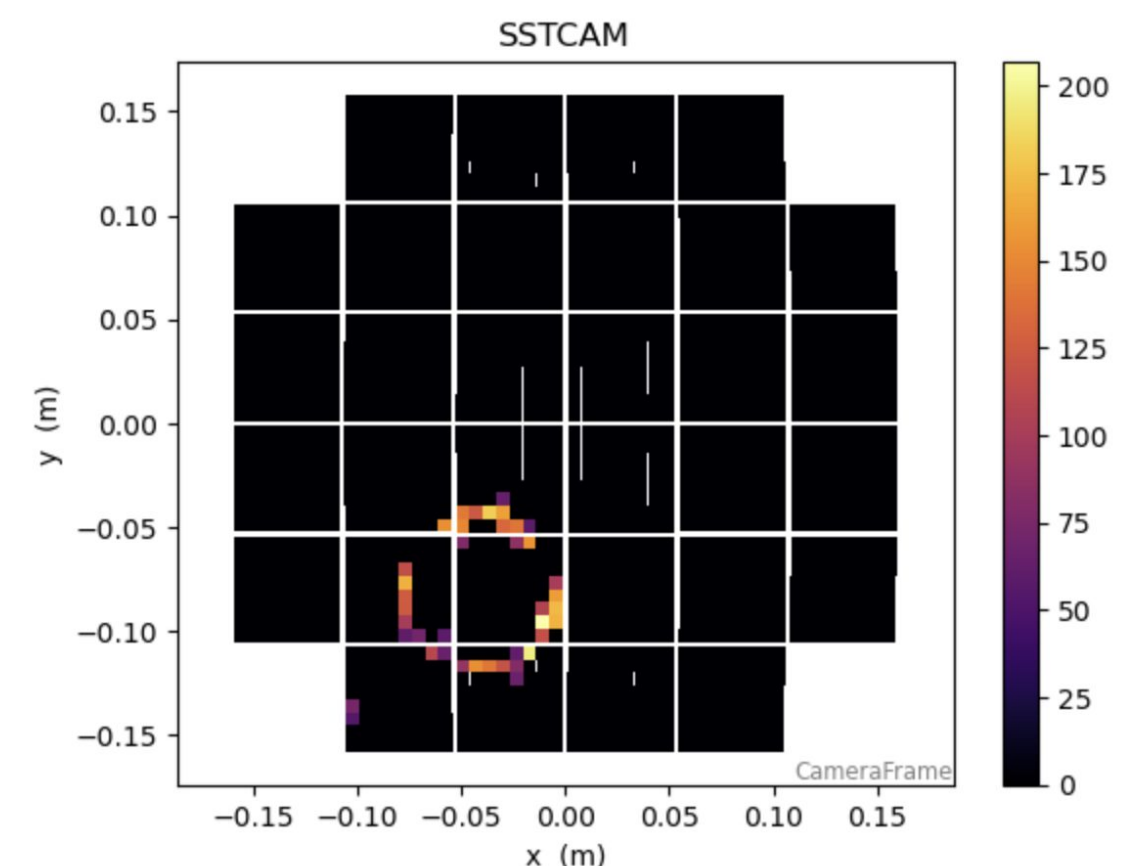
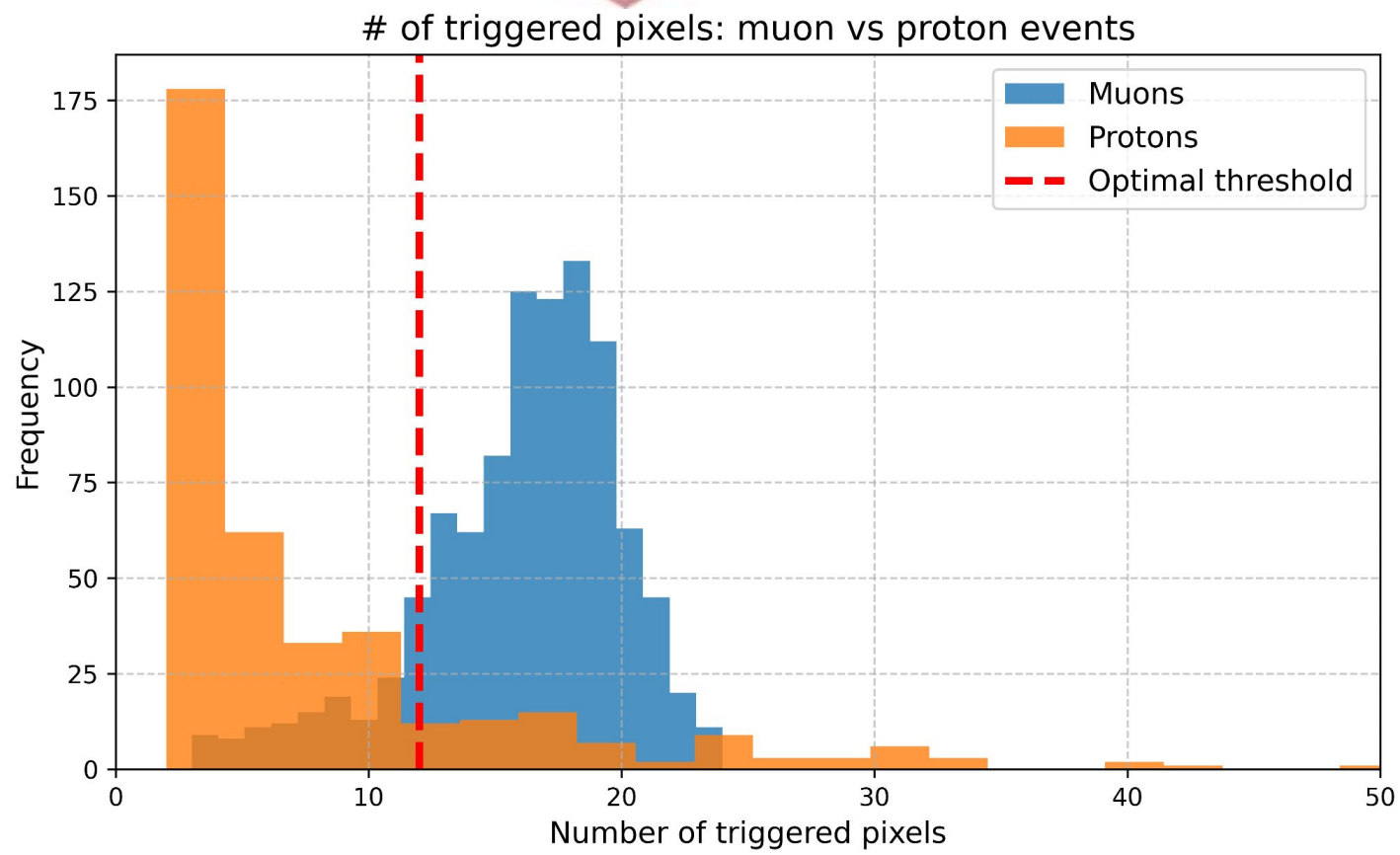
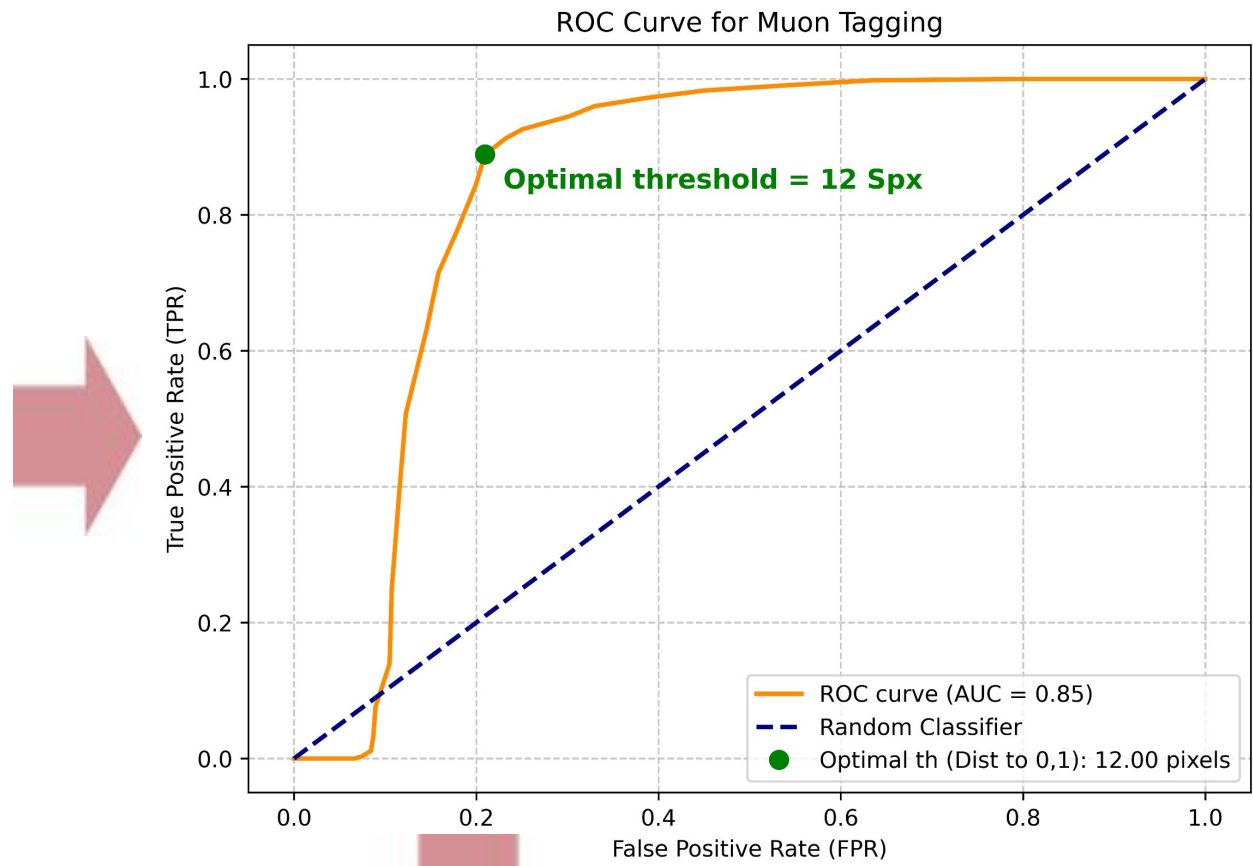
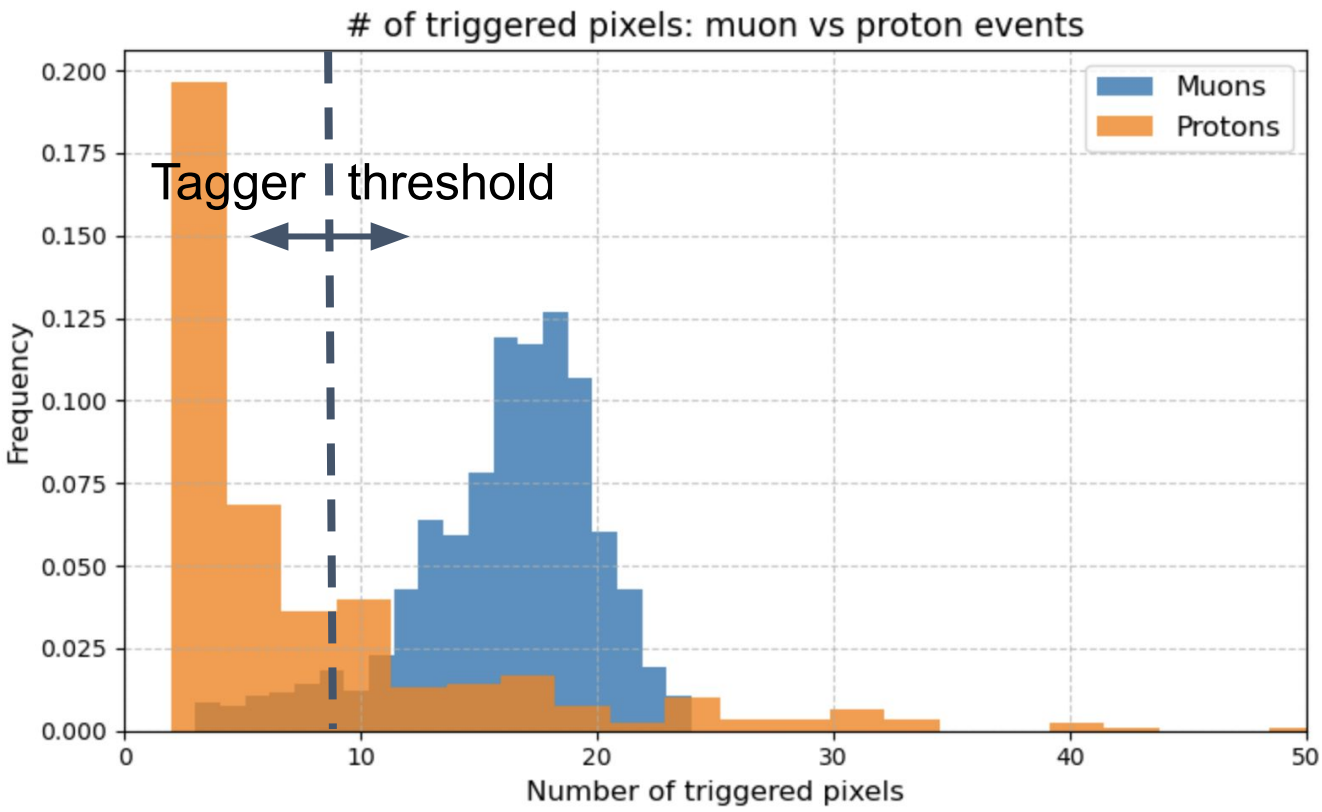
CTAO | SST
COLLABORATION

Michail Kravchenko

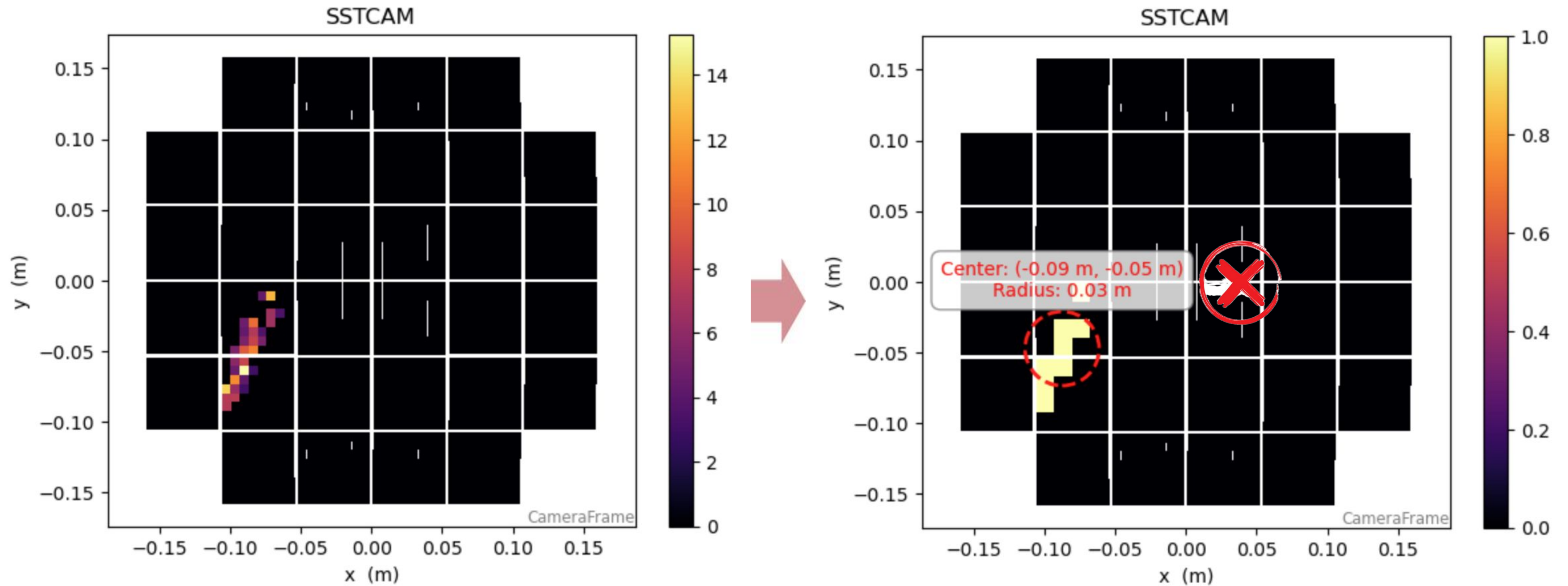
Postdoc, Gamma group

DESY, Zeuthen, Germany

Muon Tagging: Number of Triggered Pixels + Ring Completeness



Muon Tagging: Ring-Center Cleanliness Issue



Mis-tagging if # of active SP and r_{fit} are overlapped

Muon Tagging: Ring Completeness & Ring-Center Cleanness Check

- Circle fit method:

```
fit_radius, fit_x, fit_y, *_ = kundu_chaudhuri_circle_fit(
    x,
    y,
    weights
)

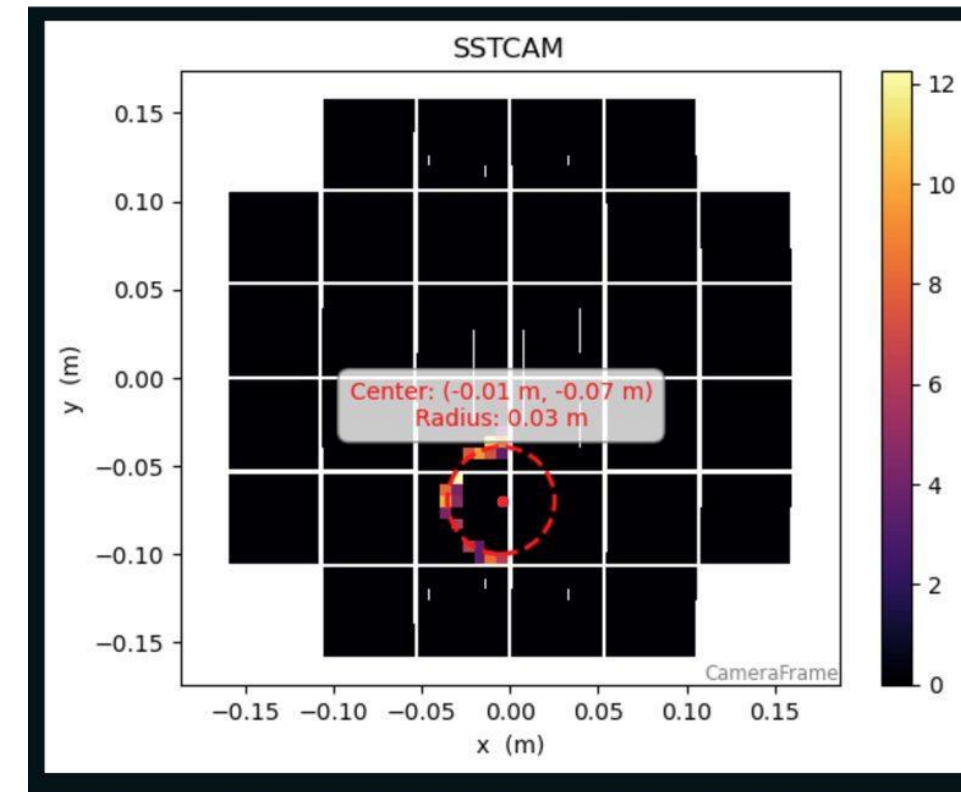
ring_frac = .65
d = np.sqrt((x - fit_x)**2 + (y - fit_y)**2)
is_pixels_at_center = (d < (ring_frac * fit_radius)) & q1[image_mask]
is_pixels_at_center = any(is_pixels_at_center)
```

ctapipe built-in

ring_completeness

+

```
ctapipe.image.muon.ring_completeness(pixel_fov_lon: Quantity,
pixel_fov_lat: Quantity, weights: Quantity | ndarray | Sequence[float],
ring: MuonRingContainer, threshold: float = 30, bins: int = 30) → float
```



Check distance from
Circle's geometric center



```
if (6 < len(trig_sp) < 20) and (not is_pixels_at_center_c):
```

Muon Tagging: Ring Completeness & Ring-Center Cleanness Check

- Circle fit method:

```
fit_radius, fit_x, fit_y, *_ = kundu_chaudhuri_circle_fit(
    x,
    y,
    weights
)

ring_frac = .65
d = np.sqrt((x - fit_x)**2 + (y - fit_y)**2)
is_pixels_at_center = (d < (ring_frac * fit_radius)) & q1[image_mask]
is_pixels_at_center = any(is_pixels_at_center)
```

ctapipe built-in

ring_completeness

+

```
ctapipe.image.muon.ring_completeness(pixel_fov_lon: Quantity,
    pixel_fov_lat: Quantity, weights: Quantity | ndarray | Sequence[float],
    ring: MuonRingContainer, threshold: float = 30, bins: int = 30) -> float
```

- Centroid depletion check method:

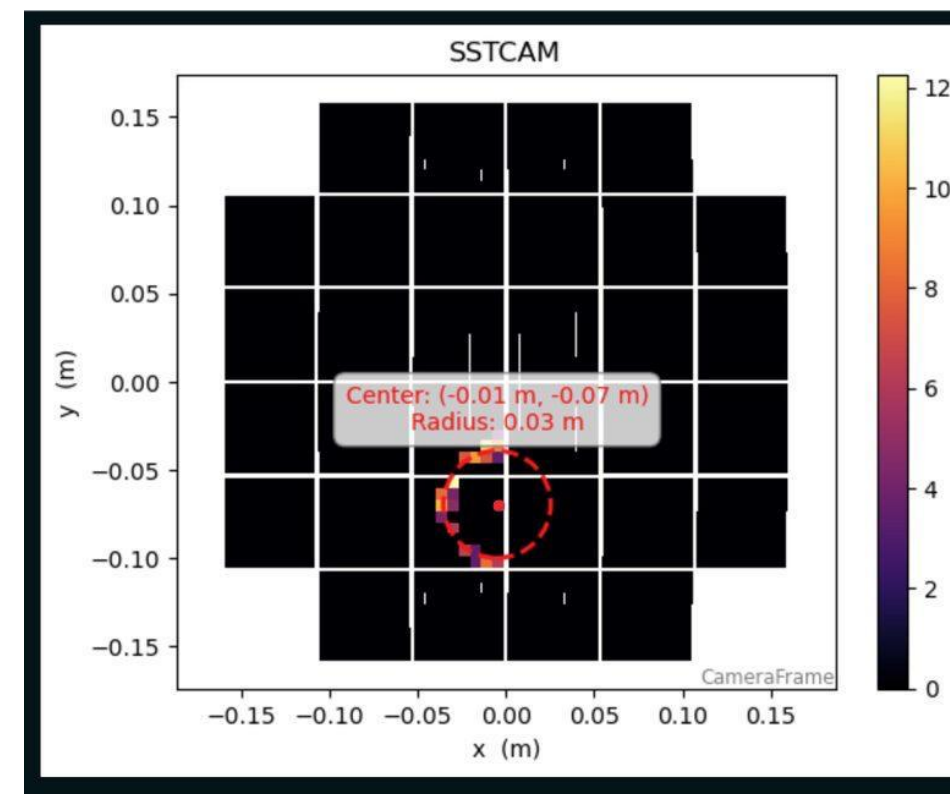
```
# Centroid
xc, yc = np.mean(x), np.mean(y)

# Euclidean distance from each pixel to the centroid
d = np.sqrt((x - xc)**2 + (y - yc)**2)

# Central exclusion zone
exclusion_zone = 0.4 * np.max(d)

is_pixels_at_center_c = np.any(d < exclusion_zone)
```

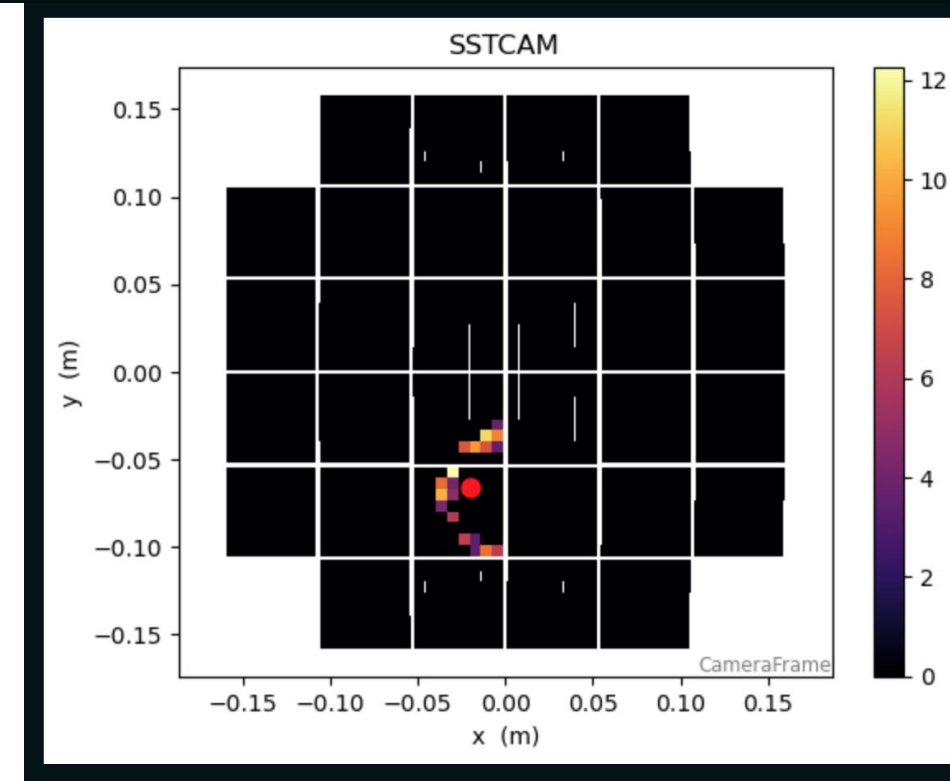
computationally simpler



Check distance from Circle's geometric center



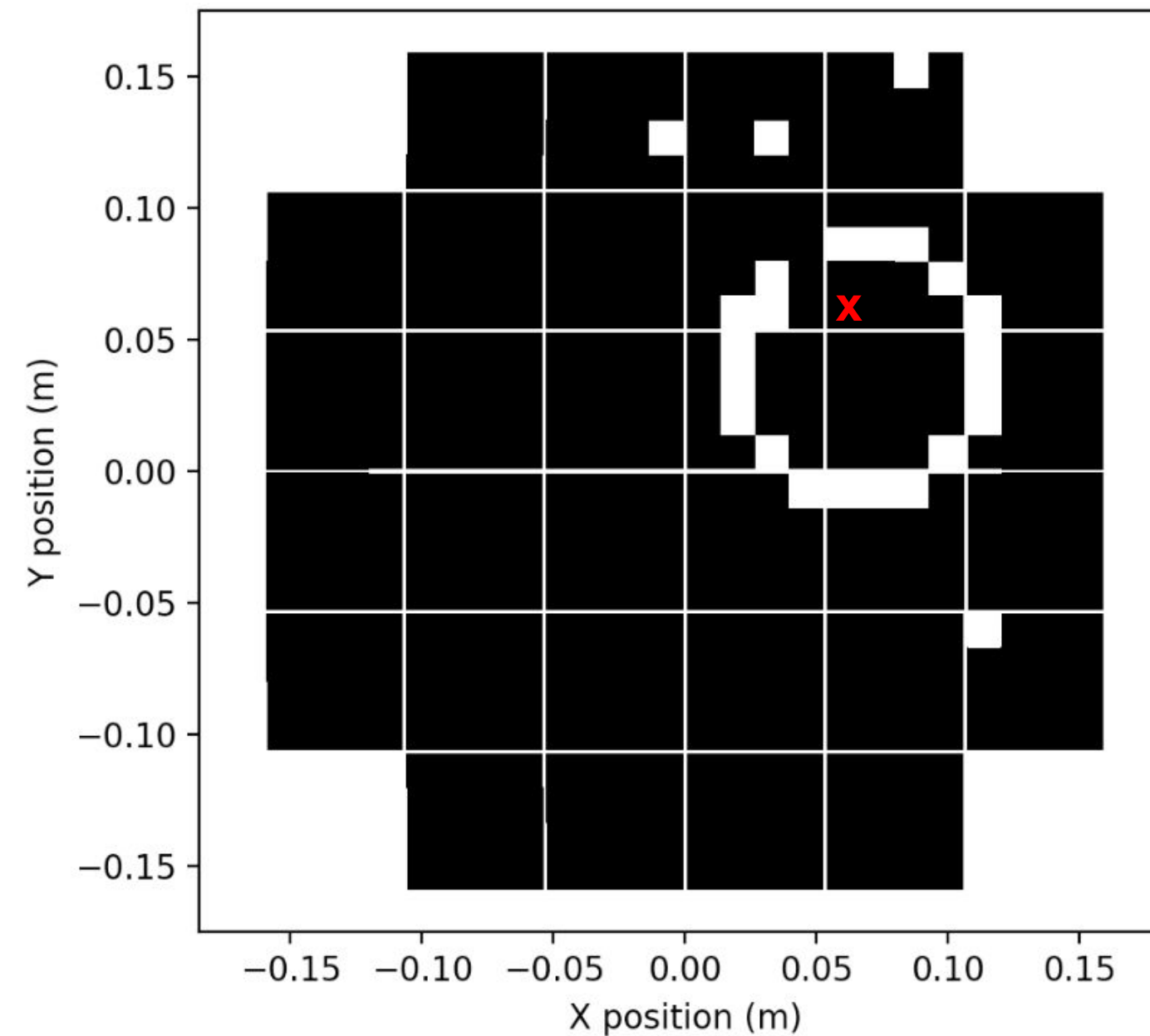
```
if (6 < len(trig_sp) < 20) and (not is_pixels_at_center_c):
```



Check distance from Pixels' center of mass



Muon Tagging: Issue w/ Centroid Depletion Check @ High-NSB



**CoM calculation is highly sensitive to isolated noise pixels
=> significant spatial bias**

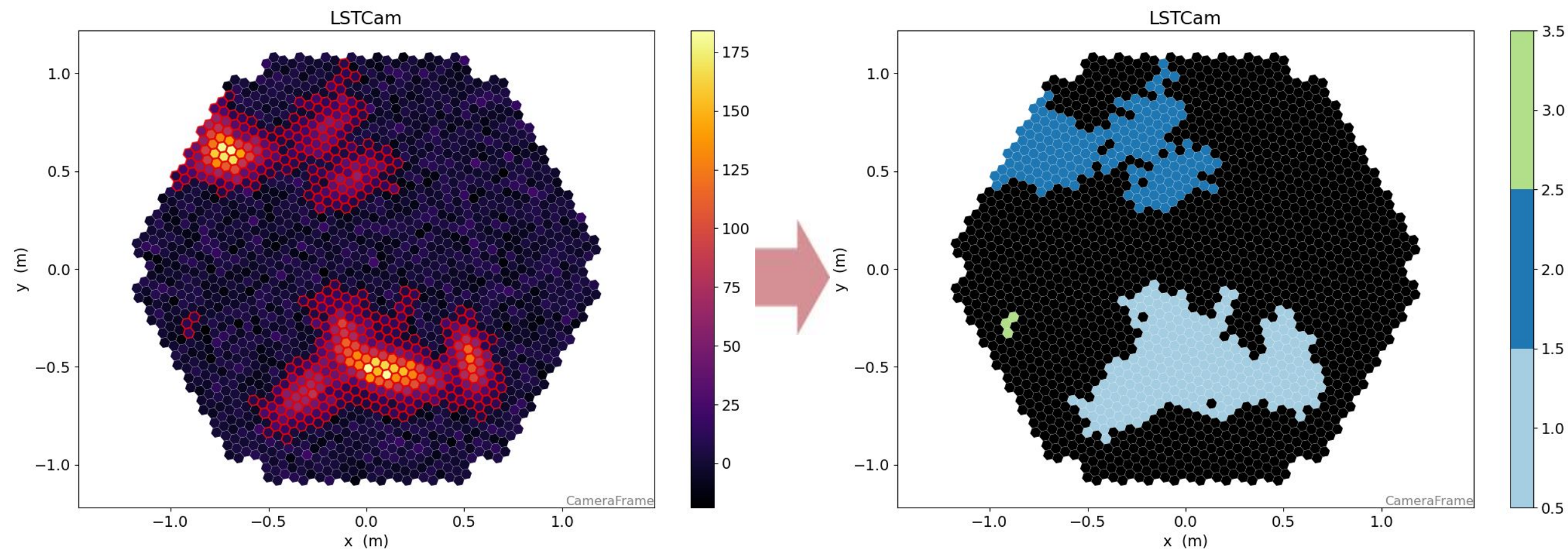
New Improved Tagging Technique

1. Using true image for finding “ground truth” rings (based on 80% completeness) all NSB levels => no need in cleaning:

```
true_image = event.simulation.tel[tel_id].true_image
```

2. def num_islands_python(camera, [clean](#)):

"""A breadth first search to find connected islands of neighboring pixels in the cleaning set"""

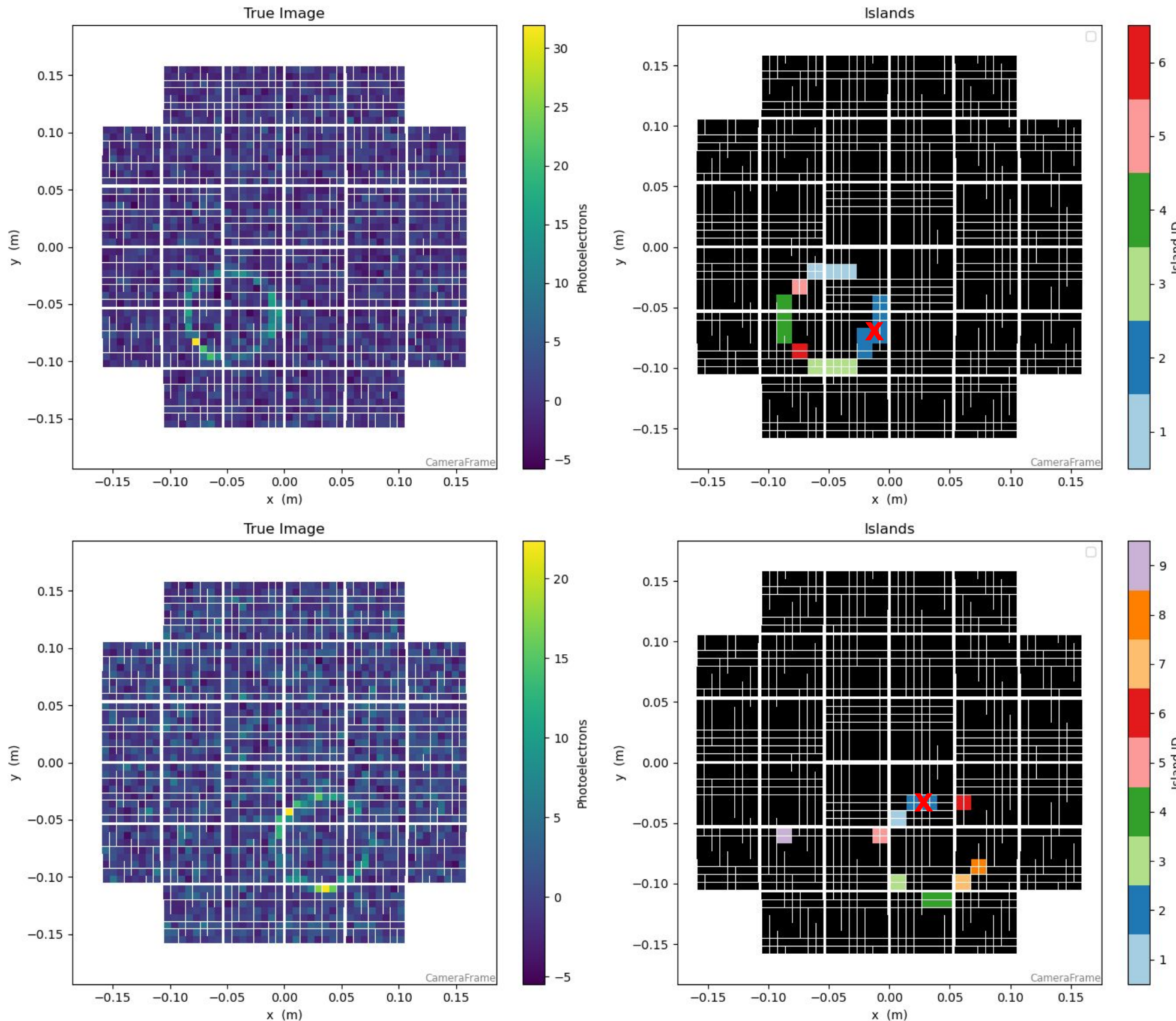


3. Calculating exclusion zone for centroid depletion check for the largest island (ring)
4. Checking exclusion zone requirement for finding pixels inside ring for all triggered SPs (including NSB/noise)

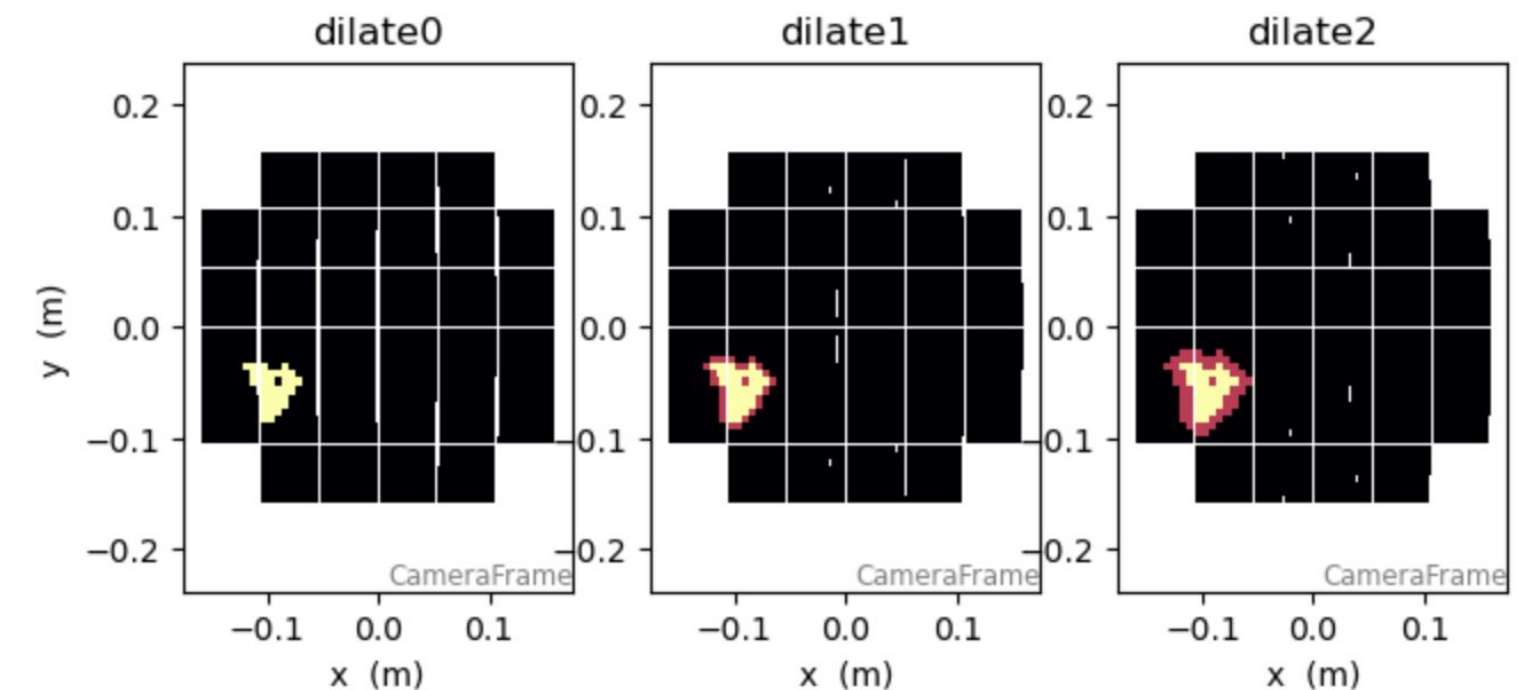
* Opt.: allowing to have some SPs inside the ring, for which the requirement above was failed

of Islands: Issue with μ Rings

Issue: Island Fragmentation



- The ctapipe built-in ``number_of_islands`` method relies on **strict nearest-neighbor connectivity**
- This method fails on segmented μ rings finding a set of fragmented islands
- **Solution: 1-2x dilation** to morphologically bridge the nearest islands => **recovering a muon ring as a single coherent island**



`dilated\_number\_of\_islands`: Implementation

```
def number_of_islands(geom, mask):
    """
    Search a given pixel mask for connected clusters.
    This can be used to separate between gamma and hadronic showers.

    Parameters
    -----
    geom : `~ctape.instrument.CameraGeometry`
        Camera geometry information, needs to be the full camera geometry,
        not already masked with ``mask``
    mask : ndarray
        input mask (array of booleans) of pixels surviving the cleaning

    Returns
    -----
    n_islands : int
        Total number of clusters
    island_labels : ndarray
        Contains cluster membership of each pixel.
        Dimension equals input geometry.
        Entries range from 0 (not in the pixel mask) to n_islands.
    """
    if geom.n_pixels != len(mask):
        raise ValueError(
            "CameraGeometry has less pixels than mask"
            ", number_of_islands needs the full CameraGeometry"
        )
    neighbors = geom.neighbor_matrix_sparse
    n_islands, island_labels = _n_islands_sparse_indices(
        neighbors.indices, neighbors.indptr, mask
    )
    return n_islands, island_labels
```

ctape built-in



```
def dilated_number_of_islands(geom, clean_mask, n_dilations=2):
    """
    Connects separated clusters by temporarily dilating the cleaned mask,
    labeling the merged components, and projecting labels back to the original pixels.
    """
    island_labels = np.zeros(geom.n_pixels, dtype=int)
    if not np.any(clean_mask):
        return 0, island_labels

    # 1. Morphological dilation to bridge physical/NSB gaps
    dilated_mask = clean_mask.copy()
    for _ in range(n_dilations):
        dilated_mask = dilate(geom, dilated_mask)

    # 2. Find connected components on the dilated camera graph
    sub_adj = geom.neighbor_matrix_sparse[dilated_mask][:, dilated_mask]
    n_dilated_islands, sub_labels = csgraph.connected_components(
        sub_adj, directed=False
    )

    # 3. Assign labels to dilated pixels
    full_dilated_labels = np.zeros(geom.n_pixels, dtype=int)
    full_dilated_labels[dilated_mask] = sub_labels + 1

    # 4. Map labels back strictly to original clean_mask pixels
    island_labels[clean_mask] = full_dilated_labels[clean_mask]

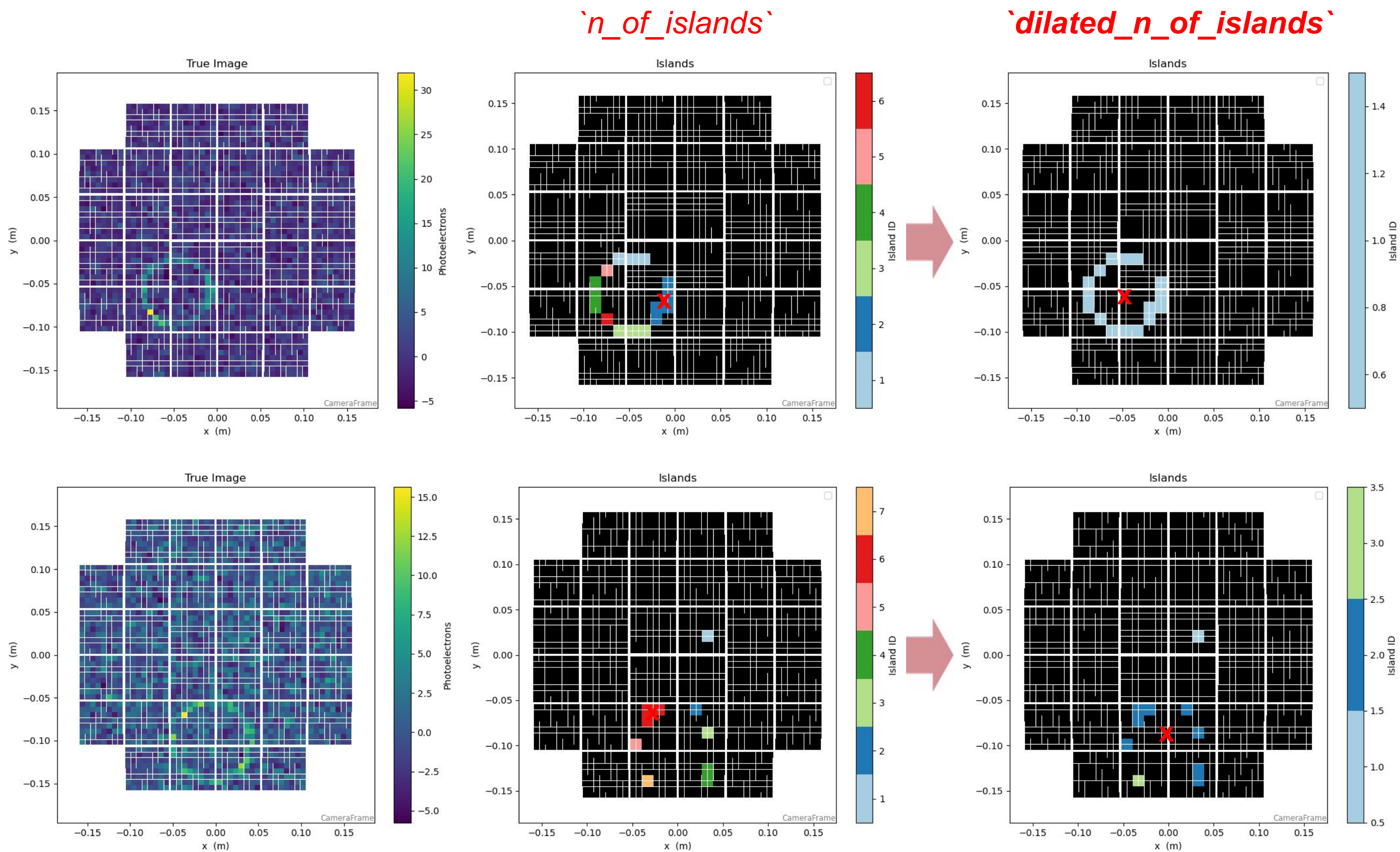
    # Re-index unique labels consecutively from 1 to N
    unique_labels = np.unique(island_labels[clean_mask])
    label_map = {old: new for new, old in enumerate(unique_labels, start=1)}

    final_labels = np.zeros_like(island_labels)
    for old, new in label_map.items():
        final_labels[island_labels == old] = new

    return len(unique_labels), final_labels
```

new advanced approach

`dilated\_number\_of\_islands`: Performance with μ Rings

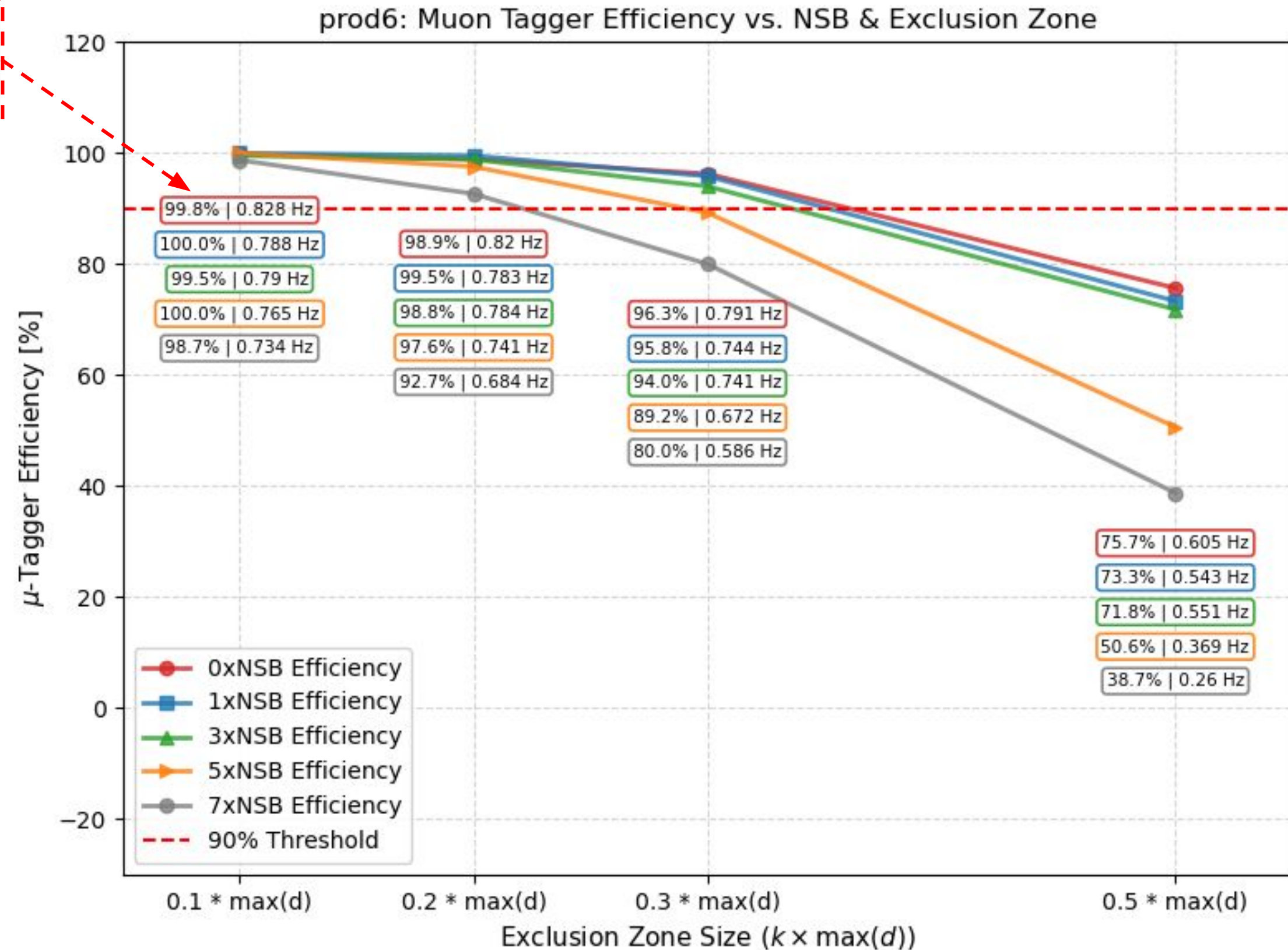


New Results: `dilated\_number\_of\_islands`

SP threshold: 10 p.e.

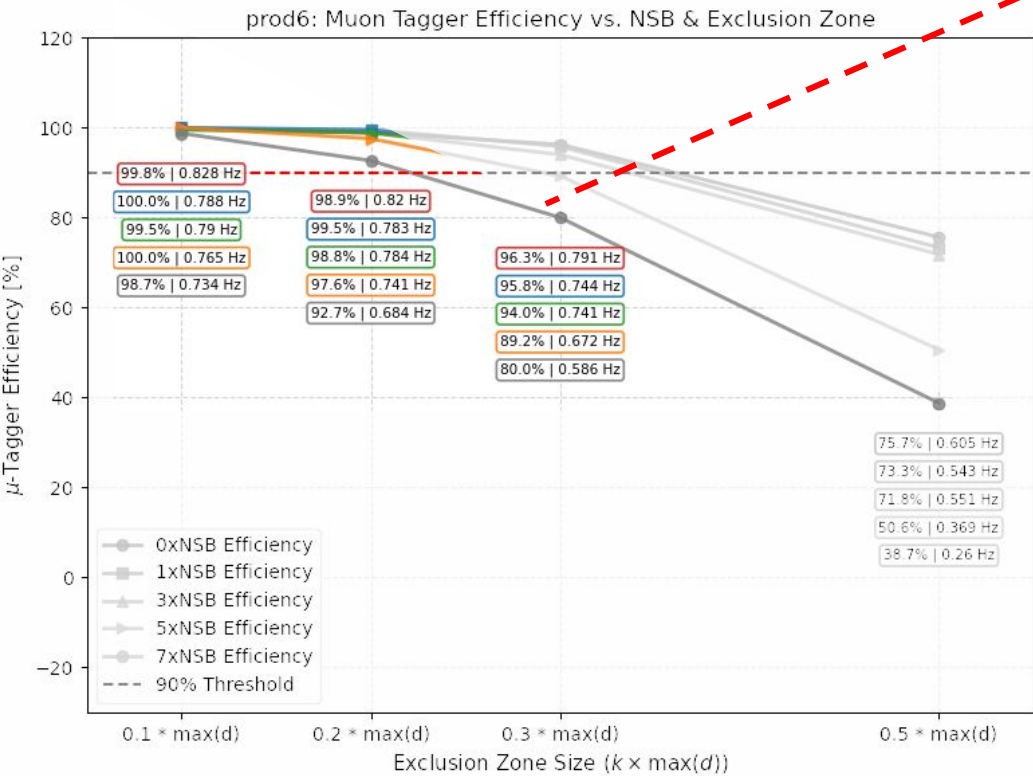
$$\text{Tagger_eff} = \frac{N_{[\text{centr_depl} \ \& \ \text{complet} > 0.8]}}{N_{[\text{complet} > 0.8]}}$$

“Ground truths” found
with `true\_image`



New Results: `dilated\_number\_of\_islands`

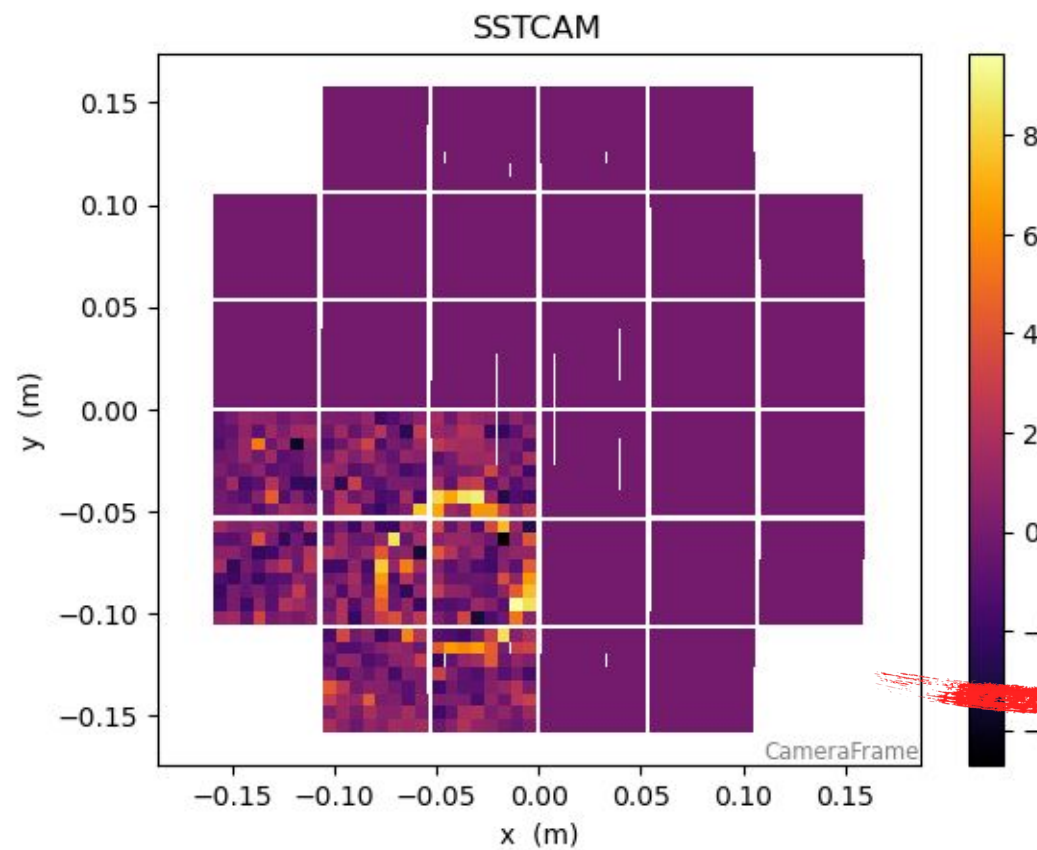
SP threshold: 10 p.e.



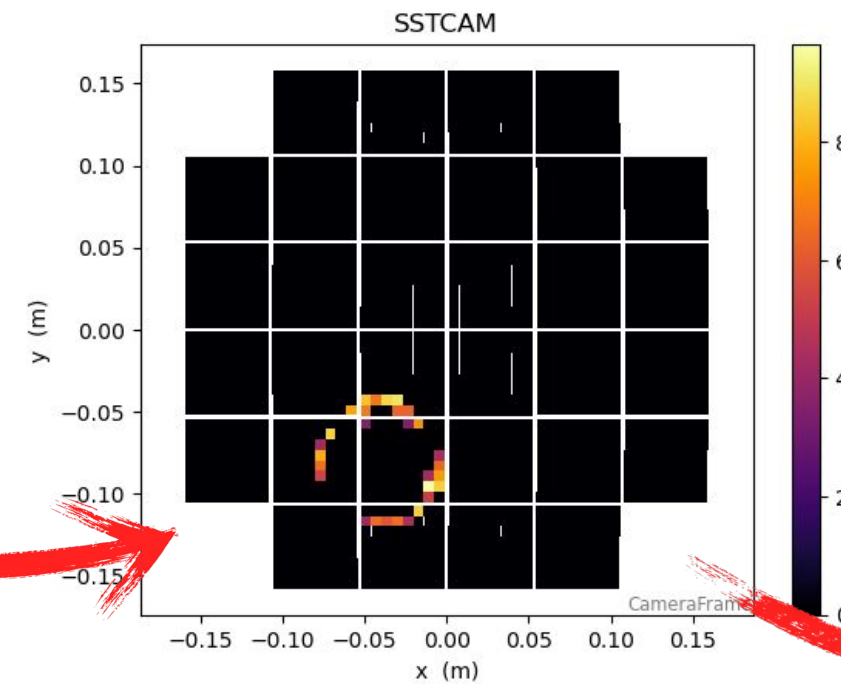
NSB & EXCLUSION ZONE SCAN SUMMARY										
NSB	Excl Zone	Showers	Detected	R_inj [Hz]	R_det [Hz]	Tel Eff [%]	Trig Eff [%]	80%-compl [%]		
0xNSB	0.10	3000	1434	1.7375	0.8280	47.6571	99.7801	50.3328		
0xNSB	0.20	3000	1420	1.7375	0.8200	47.1933	98.9003	60.4662		
0xNSB	0.30	3000	1367	1.7375	0.7908	45.5168	96.2610	81.6542		
0xNSB	0.50	3000	1041	1.7375	0.6046	34.7942	75.6598	98.7560		
1xNSB	0.10	1000	450	1.7375	0.7879	45.3455	100.0000	48.9773		
1xNSB	0.20	1000	448	1.7375	0.7835	45.0924	99.5360	59.3361		
1xNSB	0.30	1000	425	1.7375	0.7442	42.8338	95.8237	80.8219		
1xNSB	0.50	1000	317	1.7375	0.5429	31.2461	73.3179	98.7500		
3xNSB	0.10	1000	455	1.7375	0.7898	45.4551	99.5381	48.2643		
3xNSB	0.20	1000	452	1.7375	0.7845	45.1477	98.8453	60.1969		
3xNSB	0.30	1000	427	1.7375	0.7408	42.6354	93.9954	79.3372		
3xNSB	0.50	1000	315	1.7376	0.5511	31.7181	71.8245	96.8847		
5xNSB	0.10	1000	430	1.7375	0.7646	44.0047	100.0000	45.1435		
5xNSB	0.20	1000	417	1.7375	0.7415	42.6769	97.5550	57.4928		
5xNSB	0.30	1000	377	1.7375	0.6722	38.6883	89.2421	77.4947		
5xNSB	0.50	1000	209	1.7375	0.3692	21.2486	50.6112	97.1831		
7xNSB	0.10	1000	410	1.7374	0.7335	42.2199	98.7342	46.0993		
7xNSB	0.20	1000	383	1.7374	0.6839	39.3612	92.6582	55.8779		
7xNSB	0.30	1000	330	1.7374	0.5855	33.7012	80.0000	72.3112		
7xNSB	0.50	1000	155	1.7375	0.2599	14.9579	38.7342	90.0000		

MC w/ Injected Realistic Noise: Ten Noise `Bank`

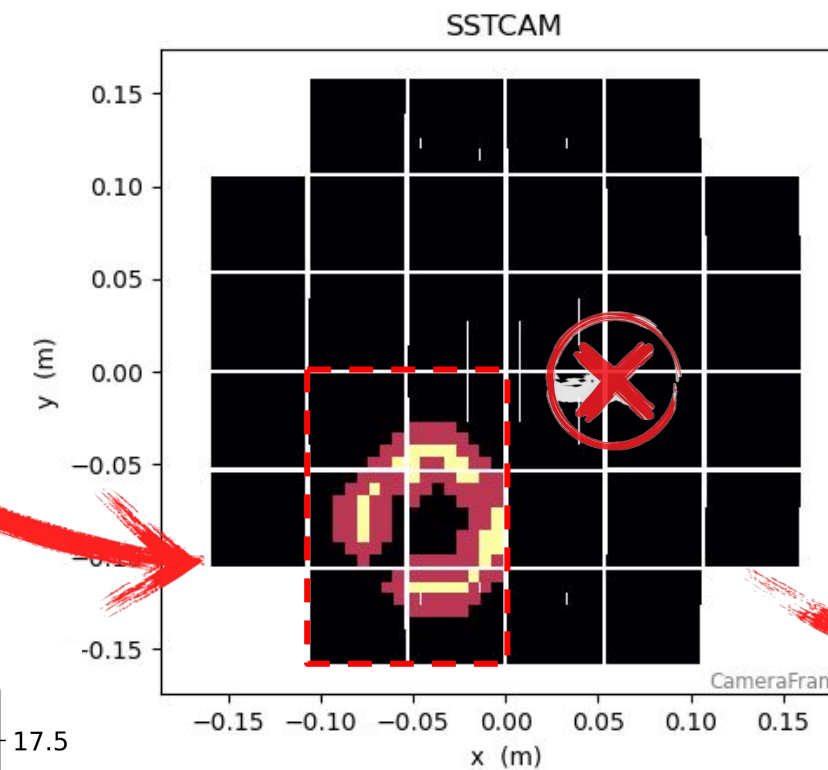
1. Image extraction & amplitude conversion (p.e.)



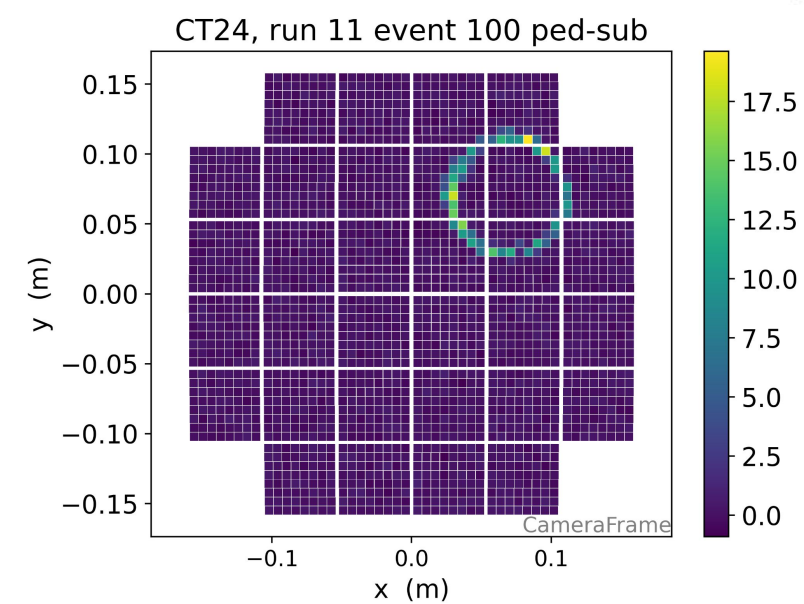
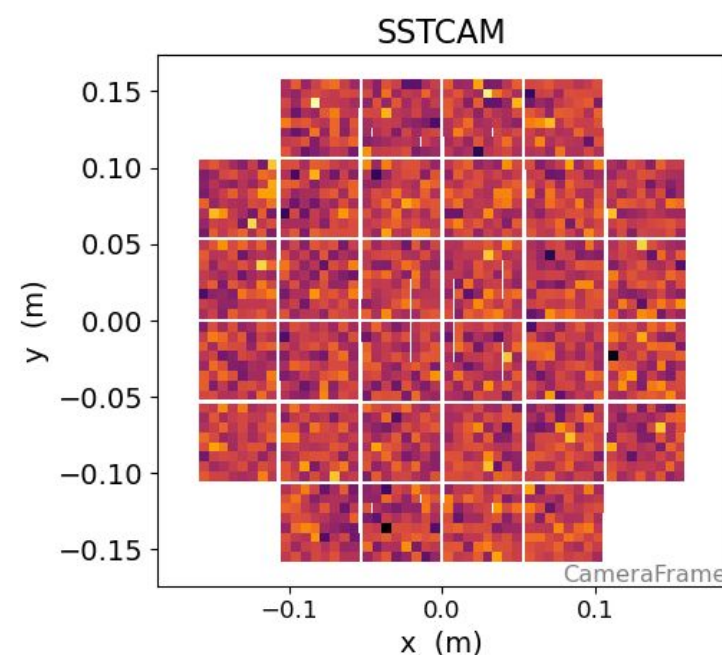
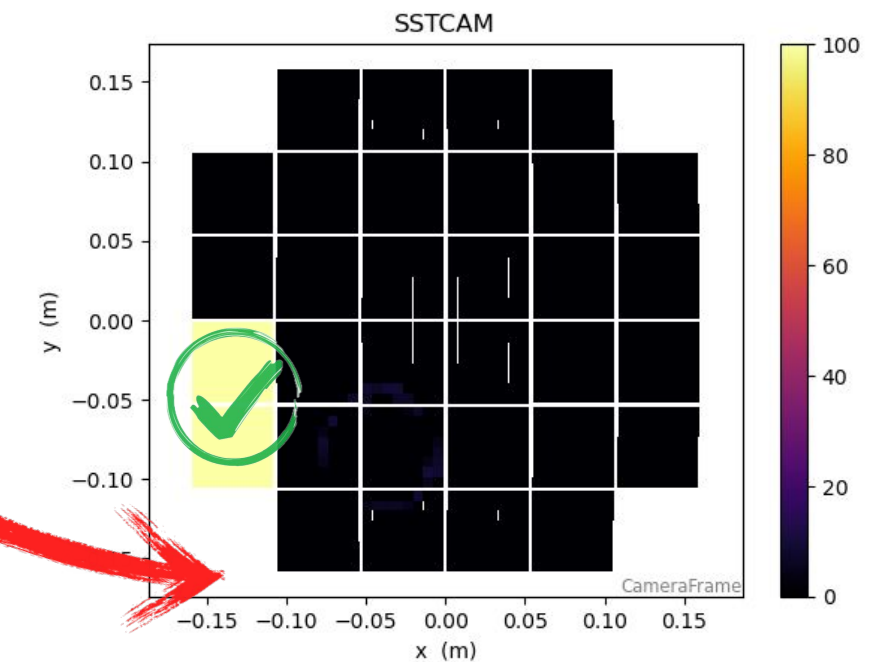
2. Image cleaning



3. Dilation



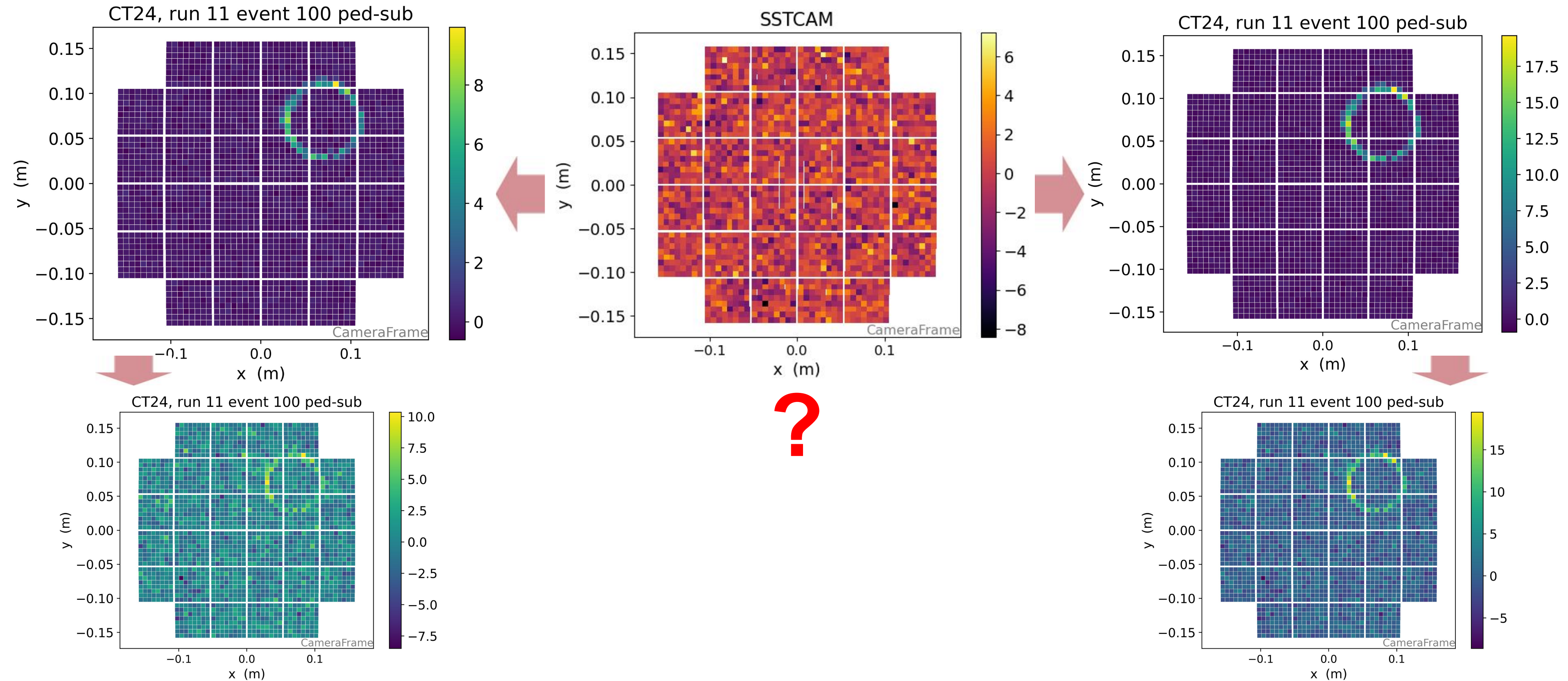
4. Storing TMs w/o signal



MC and Ten Data: Amplitude Matching Issue

$[r0_wf - ped] / fadc_amplitude$ (6.77)

$r1_wf$



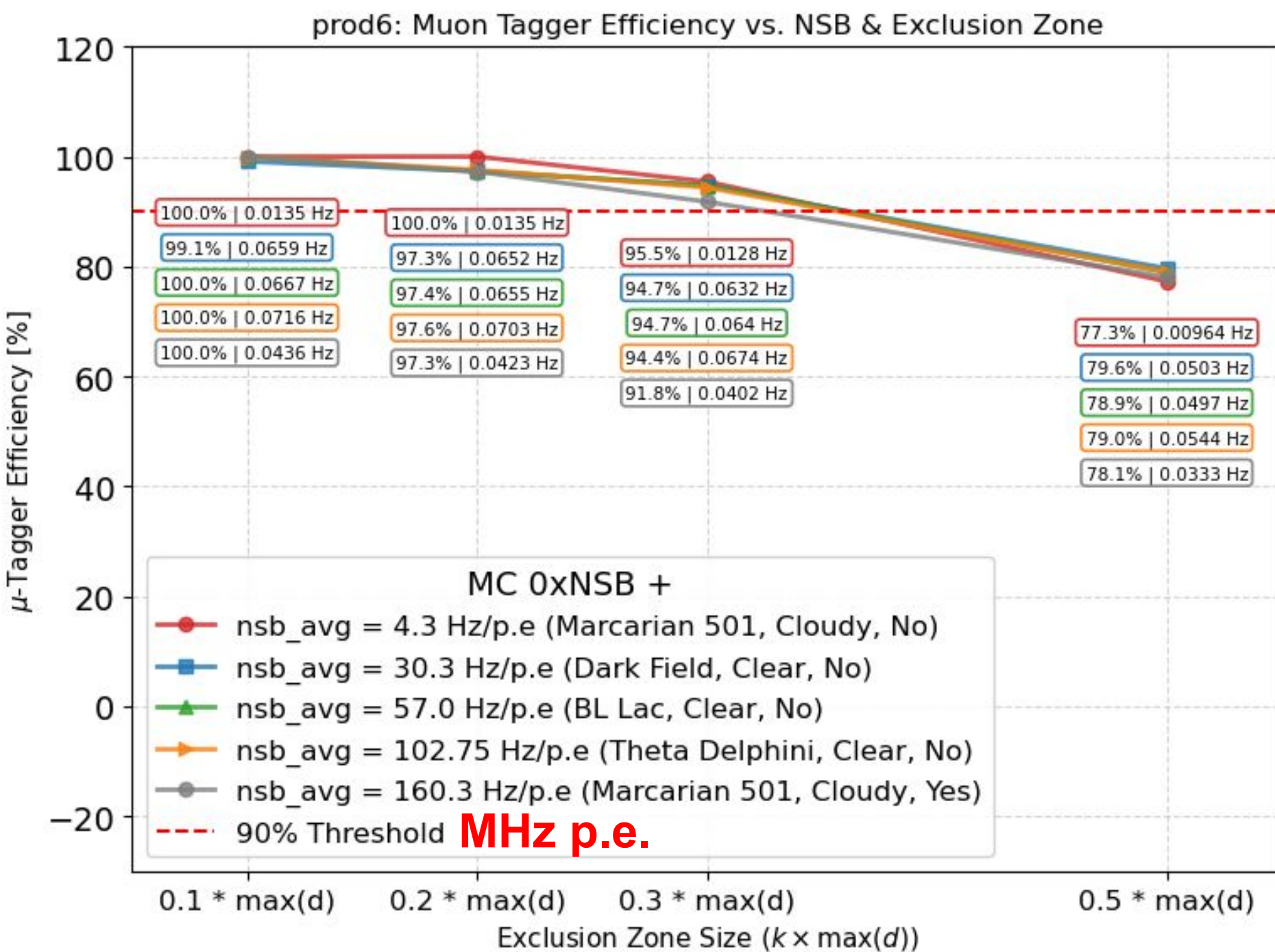
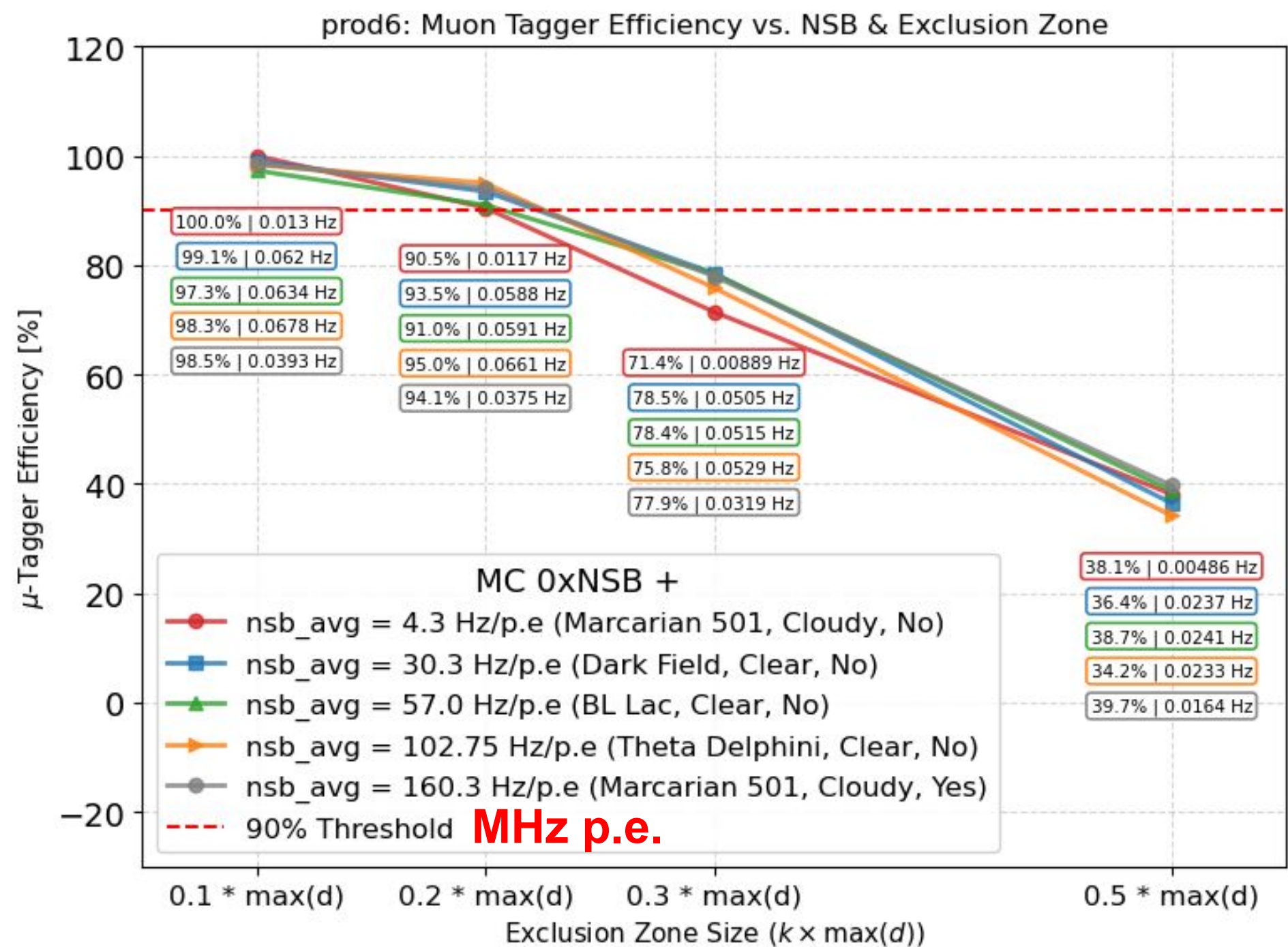
New Results: MC 0xNSB + Injected Ten Noise

[VERY PRELIMINARY RESULTS]

MC_0xNSB: r1_wf

SP threshold: 15 p.e.

SP threshold: 10 p.e.



Overview

Improved μ tagging is converging to a final optimal approach:

- Tagging based on # of triggered SPs
- + Ring-center cleanness and ring-completeness check based on centroid depletion method
- + Newly implemented `dilated\_number\_of\_islands` method to isolate muon ring from NSB\noisy SPs

Next Steps/TODOs

- Double-check of the results, including MC data with half/full moon
- Further NSB scan efficiency with injected Tenerife noise and comparison to MC data: is NSB under-/overestimated in current MC simulations?
- Optimizing tagging parameters with separate SP thresholds for low- and high NSBs

Backup

A fast muon tagger for CHEC-S

A camera prototype for the Small Sized Telescopes of
CTA

Roberta Pillera

Università degli Studi di Bari "Aldo Moro"

3.2 Night sky background

The events used for testing in Sec. 3.1 have been simulated at a low night sky background (NSB) level corresponding to observations of the extragalactic region of the sky. Twice that level (medium NSB) corresponds to the galactic plane region, whereas with 3 times NSB (high NSB) to the galactic center. With higher NSB the trigger thresholds are adapted. This has not been included

Impact distance/parameter

